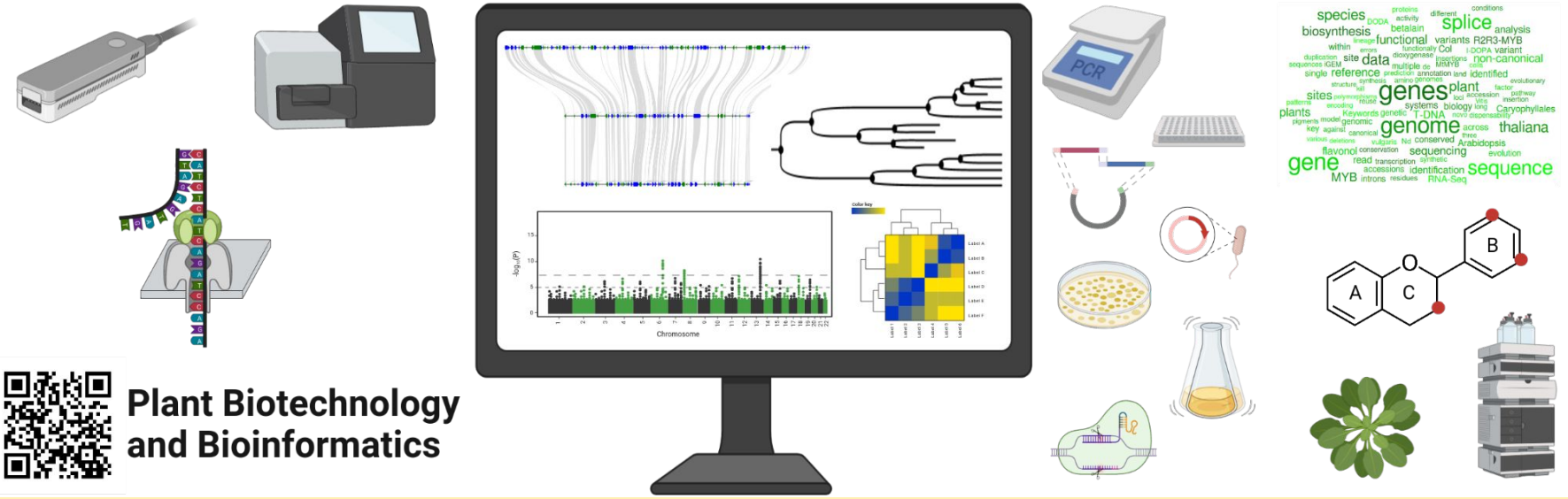
Technische Universität Braunschweig



Plant Biotechnology and Bioinformatics

Python - File handling

Prof. Dr. Boas Pucker (Plant Biotechnology and Bioinformatics)

Availability of slides

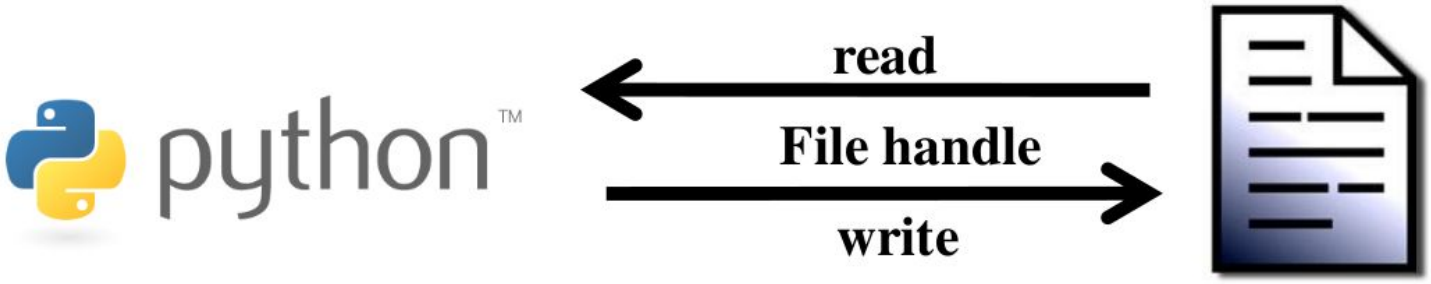
- All materials are freely available (CC BY) - after the lectures:
  - StudIP: ‘Python for Life Scientists’
  - GitHub: https://github.com/bpucker/teaching

- Questions: Feel free to ask at any time

- Feedback, comments, or questions: b.pucker[a]tu-bs.de

Concept of file handling

- ‘Connection’ from Python to file
- Read = transfer of data from file
- Write = transfer of data into file



Reading a file (parsing)

```
f = open("test.txt", "r") #this opens a connection between Python and file
#f = connection from Python to file (file handle)
#'r' (reading) is default
lines = f.readlines() #reads content of file into a list
#each line becomes a list element of 'lines'
f.close() #close connection between Python and file
```

```
with open("test.txt", "r") as f: #opens file to read
    lines = f.readlines()
```

```
with open("test.txt", "r") as f: #opens file to read
    content = f.read()   #reads entire file content into one string
```

Reading a file (big data)

- Advantage: only one line is read and processed at a time
- NGS data (e.g. FASTQ/SAM/BAM/VCF) are usually several GB in size
  - might exceed RAM
- Very long sequence (e.g. genome sequences) in FASTA
  - might be to large for available RAM

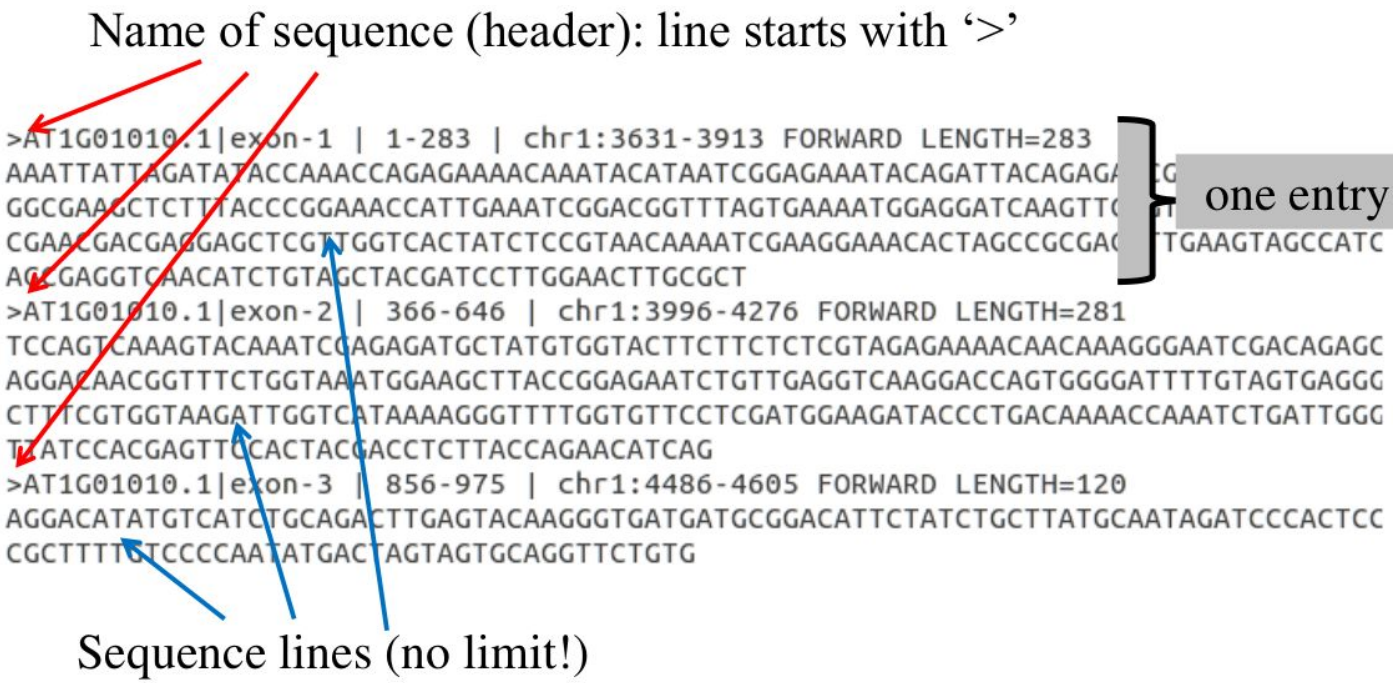
```
with open("text.txt", "r") as f: #"r" is default
    #f = connection from Python to file (file handle)
    line = f.readline() #reads next line
    while line: #until end of file is reached
        print(line)
        line = f.readline() #reads next line
```

File analysis - example

- How many lines are in AtCol0_Exons.fasta?
- Linux function: $ head <FILENAME>

```
>AT1G01010.1|exon-1 | 1-283 | chr1:3631-3913 FORWARD LENGTH=283
AAATTATTAGATATACCAAACCAGAGAAAACAAATACATAATCGGAGAAATACAGATTACAGAGAGCGAGAGAGATCGAC
GGCGAAGCTCTTTACCCGGAAACCATTGAAATCGGACGGTTTAGTGAAAATGGAGGATCAAGTTGGGTTTGGGTTCCGTC
CGAACGACGAGGAGCTCGTTGGTCACTATCTCCGTAACAAAATCGAAGGAAACACTAGCCGCGACGTTGAAGTAGCCATC
AGCGAGGTCAACATCTGTAGCTACGATCCTTGGAACTTGCGCT
>AT1G01010.1|exon-2 | 366-646 | chr1:3996-4276 FORWARD LENGTH=281
TCCAGTCAAAGTACAAATCGAGAGATGCTATGTGGTACTTCTTCTCTCGTAGAGAAAACAACAAAGGGAATCGACAGAGC
AGGACAACGGTTTCTGGTAAATGGAAGCTTACCGGAGAATCTGTTGAGGTCAAGGACCAGTGGGGATTTTGTAGTGAGGG
CTTTCGTGGTAAGATTGGTCATAAAAGGGTTTTGGTGTTCCTCGATGGAAGATACCCTGACAAAACCAAATCTGATTGGG
TTATCCACGAGTTCCACTACGACCTCTTACCAGAACATCAG
>AT1G01010.1|exon-3 | 856-975 | chr1:4486-4605 FORWARD LENGTH=120
AGGACATATGTCATCTGCAGACTTGAGTACAAGGGTGATGATGCGGACATTCTATCTGCTTATGCAATAGATCCCACTCC
CGCTTTTGTCCCCAATATGACTAGTAGTGCAGGTTCTGTG
```

Multiple FASTA file



Analyze FASTA file - example

```
1 with open( "/vol/apbiokurs/data/AtCol0_Exons.fasta", "r" ) as f:
2     line = f.readline() #reading first line
3     line_counter = 0   #counting lines
4     while line:
5         line_counter += 1    #counting lines
6         line = f.readline()
7 #number in line_counter needs to be converted to string:
8 print("File contains " + str( line_counter ) + " lines")
```

Exercises - Part4a

- 4.1) Count number of sequences (=number of headers) in example FASTA file!
- 4.2) Count number of sequence lines!
- 4.3) Count number of characters in document! (How many per line?)
- 4.4) How long are all contained sequences combined?
- 4.5) Calculate the average sequence length in this file!

And back again … writing into file!

- If output file does not exist, it will be created
- File handle (f and out) can have any other name

```
1   Read:
2   with open( "test.txt", "r" ) as f:  #"r" (read) ist default
3       lines = f.readlines()
4
5
6
7
8   Write:
9   with open( "test2.txt", "w" ) as out:
10      out.write( "hello world!" )
```

difference: r = read; w = write

Writes a string into a file

Read & write

```
1  with open( "test.txt", "r" ) as f:  #open file to read
2      with open( "test2.txt", "w" ) as out:   #open file to write
3          line = f.readline() #read from first file
4          while line:
5              #this would be the place to apply filters
6              out.write( line )
7              line = f.readline()
```

Exercises - Part4b

- 4.6) Read the file AtCol0_Exons.fasta and write all headers (starting with '>') into a new file!

- 4.7) Read the file AtCol0_Exons.fasta and write the following into the output file:
  - The line if it is a header
  - The length of the line if it is a sequence line

White space characters

- New line (‘\n’) and tab (‘\t’) are special characters
  - Print ‘/hello\tworld!\nhello\tworld!\n’

- Python interprets these characters in print statements, but functions like readline() and write() do not!
  - New line (‘\n’) needs to be added “manually” to each line

```
1 with open( "test_file.txt", "w" ) as out:
2     out.write( "first test" )
3     out.write( "second test" )
4     out.write( "third test\n" )
5     out.write( "fourth test\n" )
6     out.write( "fifth test" )
7     out.write( "sixth test" )
```

How many lines does this file have?

strip()

- Removes white space characters from borders of a string
- often used to remove new lines (‘\n’) at the line end

```
1  line = ">name_of_first_seq\n"
2  print( line )
3  #>name_of_first_seq
4  # [empy line genereated by \n ]
5  line = line.strip()
6  print( line )
7  #>name_of_first_seq
```

split()

- Separates a string at each given occurrence of given substring
- Frequent separators: tab, comma, …
- Generates list of strings

```
#tab-delimited file
line = "column1\tcolumn2\tcolumn3\tcolumn4\tcolumn5\n"
#line should be splitted at tabs
columns = line.strip().split('\t')
print(columns)
#["column1", "column2", "column3", "column4", "column5"]
```

join()

- Combines strings of a list by putting a given substring between them
  - Examples: tab, space, comma, underscore

- Important: all elements of the list need to be string!

```
#tab-delimited file
line = "column1\tcolumn2\tcolumn3\tcolumn4\tcolumn5\n"
#line should be splitted at tabs
columns = line.strip().split('\t')
print(columns)
#["column1", "column2", "column3", "column4", "column5"]

new_line = "_".join(columns)
print(new_line)
#column1_column2_column3_column4_column5
```

Exercises - Part4c

- 4.8) Calculate the number of sequences, the cumulative length, and the average length based on your results of exercise 4.7! Are they matching your previously calculated values?
- 4.9) Write sequences into a new file if their length is a multiple of 10!
- 4.10) How many genes are located on Chr3?
- 4.11) What is the gene density of each chromosome (genes per Mbp)?
- 4.12) Write all sequences located on Chr2 between 10 and 15 Mbp.

Exercises - Part4d

- X4.1) Print the GC content of all genes with more than 10 exons.
- X4.2) Write the number of exons per gene into a new file.
- X4.3) Read the file AtCol0_Exons.fasta and write the following:
  - Only the Arabidopsis Gene Identifier (AGI, e.g. AT1G01010)
  - Gene identifier, exon name, and exon length (tab-delimited)

```
>AT1G01010.1|exon-1 | 1-283 | chr1:3631-3913 FORWARD LENGTH=283
AAATTATTAGATATACCAAACCAGAGAAAACAAATACATAATCGGAGAAATACAGATTACAGAGAGCGAGAGAGATCGAC
GGCGAAGCTCTTTACCCGGAAACCATTGAAATCGGACGGTTTAGTGAAAATGGAGGATCAAGTTGGGTTTGGGTTCCGTC
CGAACGACGAGGAGCTCGTTGGTCACTATCTCCGTAACAAAATCGAAGGAAACACTAGCCGCGACGTTGAAGTAGCCATC
AGCGAGGTCAACATCTGTAGCTACGATCCTTGGAACTTGCGCT
>AT1G01010.1|exon-2 | 366-646 | chr1:3996-4276 FORWARD LENGTH=281
TCCAGTCAAAGTACAAATCGAGAGATGCTATGTGGTACTTCTTCTCTCGTAGAGAAAACAACAAAGGGAATCGACAGAGC
AGGACAACGGTTTCTGGTAAATGGAAGCTTACCGGAGAATCTGTTGAGGTCAAGGACCAGTGGGGATTTTGTAGTGAGGC
CTTTCGTGGTAAGATTGGTCATAAAAGGGTTTTGGTGTTCCTCGATGGAAGATACCCTGACAAAACCAAATCTGATTGGG
TTATCCACGAGTTCCACTACGACCTCTTACCAGAACATCAG
>AT1G01010.1|exon-3 | 856-975 | chr1:4486-4605 FORWARD LENGTH=120
AGGACATATGTCATCTGCAGACTTGAGTACAAGGGTGATGATGCGGACATTCTATCTGCTTATGCAATAGATCCCACTCC
CGCTTTTGTCCCCAATATGACTAGTAGTGCAGGTTCTGTG
```

Exercises - Part4d

- X4.4) Write a function that loads the entire FASTA file content.
- X4.5) Calculate the length of the shortest and longest sequence in the file.
- X4.6) Write the reverse complement of every sequence into an output file.
- X4.7) Identify systematic differences between properties of genes on the FORWARD vs. REVERSE strand (number, length, sequence composition)

Time for questions!